Course Name: Software Development Project
Course No: CSE 3106

Architecture Pattern For Zen Screens

Submitted to:

Dr. Amit Kumar Mondal
Associate Professor,
Computer Science & Engineering Discipline,
Khulna University

Submitted By:

LM Raihan Hamid
Student ID: 200218
&
MD Mehedi Hasan Jubair
Student ID: 200231

Project Title: Zen Screens,

A screen saver software.

The Architecture Pattern we determined for:

Model-View-Controller(MVC) Pattern

Based on the requirements provided, an appropriate architecture pattern for the Zen Screens screen saver software could be the Model-View-Controller (MVC) pattern. MVC separates the concerns of an application into three interconnected components: the Model, View, and Controller. This separation helps in organizing code, improving maintainability, and facilitating flexibility.

Model-View-Controller (MVC) pattern in our project:

Model:

Description: The Model represents the data and the business logic of the application.

Accountabilities:

1. Managing the configuration settings of the screen saver (e.g., duration before activation, chosen visuals).

2. Handling the logic for loading, managing, and displaying visuals (graphics, animations, images).

3. Managing the interaction with the operating system to detect idle time and activate the screen saver accordingly.

4. Ensuring data integrity and consistency.

The Model component is independent of the user interface and interacts with both the View and Controller components.

View:

Description: The View represents the presentation layer of the application. It is responsible for rendering the user interface elements and displaying information to the user.

Accountabilities:

1. Rendering the configuration settings interface where users can customize the visuals of the screen saver.
2. Providing a preview of the selected visuals to the user.
3. Displaying user documentation and troubleshooting information.

The View component interacts with the Model to fetch data to be displayed and with the Controller to handle user input and actions.

Controller:

Description: The Controller acts as an intermediary between the Model and the View. It handles user input, updates the Model accordingly, and triggers changes in the View.

Accountabilities:

1. Capturing user input from the interface (e.g., configuration changes, activating the screen saver).

2. Updating the Model based on user input and interactions.

3. Triggering the display of the screen saver when the computer is idle for a specified duration.

The Controller component communicates with both the Model and the View, ensuring that they stay decoupled and independent of each other.

Graphical Representation of MVC pattern:

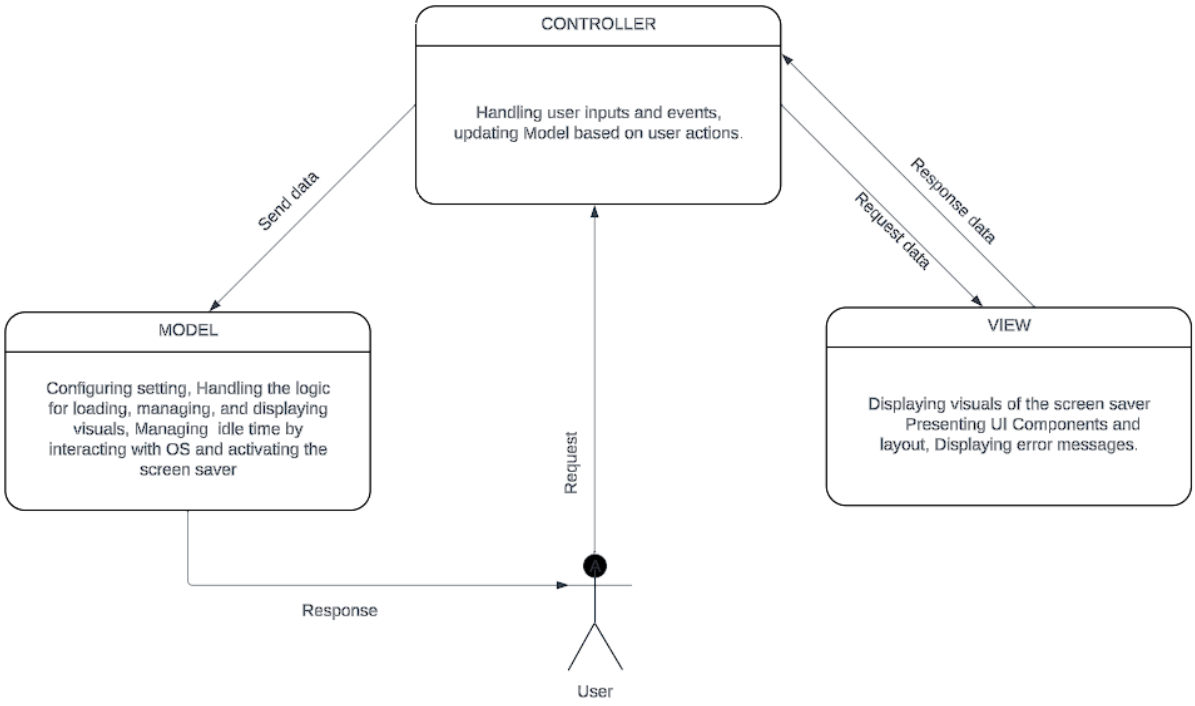

Fig: Diagram of MVC in Zen Screens

(https://lucid.app/lucidchart/71448e09-41db-45c7-a40a-3cd3f6c46759/edit?viewport_loc=-105%2C306%2C2742%2C1330%2C0_0&invitationId=inv_8f6c52c8-29a2-462e-b3e9-0f3247ee53a2)